

=====
Apr.15. 1985

Vol.1, No.10

\$1.60
=====

Special Announcement

Our guest speaker this month will be Jim Butterfield, from Toronto. Well known, especially by Commodore users, Mr. Butterfield's talk is expected to be interesting to all. See the details inside this issue.

INSIDE ...

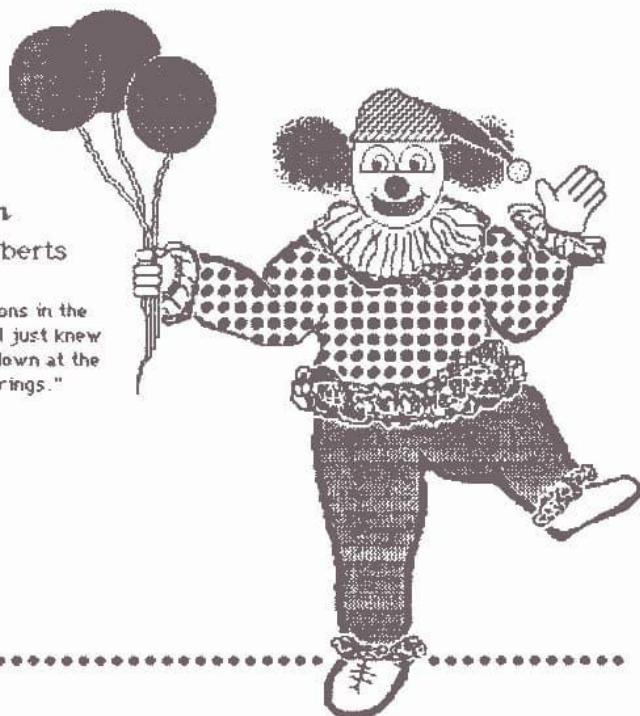
Did You Miss	4
About Jim Butterfield	4
April Agenda	5
Adams Apple Notes	6
Strings in Comal	6
Come Forth!	8
C-64 Games Design	9
Book Review	11
A Pro Looks at ProDOS	12
March Minutes	14

Mac T. Knife

Funny Clown

by Karen McRoberts
Evansville, IN

"When I saw the balloons in the
MacPaint™ manual, I just knew
they had to have a clown at the
other end of their strings."



RENEWAL TIME

NAME _____

ADDRESS _____

CITY _____

PROV. _____

POSTAL CODE _____

PHONE _____

COMPUTER _____

OHC # _____

PLEASE MAKE CHEQUES PAYABLE TO: (\$15/YEAR)

OTTAWA HOME COMPUTING

P.O. BOX 4164, STATION 'C', OTTAWA, ONTARIO, K1Y 4P3

OTTAWA HOME COMPUTING

OTTAWA HOME COMPUTING is the newsletter of the Ottawa Home Computer Club. Membership is open to all with a genuine interest in personal computing for \$15/year in Canada. Membership includes OTTAWA HOME COMPUTING. Meetings are usually held on the third Monday of each month, 7:30 PM, at Charlebois High School, corner of Heron Road and Alta Vista Drive in Ottawa.

When submitting articles please print or type with a fresh ribbon, 8-1/2 x 11 inch white paper, double-spaced on one side only. Leave one-and-a-half inch borders on all four sides.

Permission to reprint articles appearing herein is hereby granted, provided credit to both the author and Ottawa Home Computing is given.

Advertisements of personal items up to 25 words for one insertion per issue are free to members. Inquire for commercial rates.

President	
Brian Morrow	829-8818
Treasurer	
Brent Goss	824-8571
Membership	
Mike Bryan	226-5588
Bulletin Board Editor	
Don White	829-2082
Newsletter Editor	
Paul Irwin	728-6728
Apple Librarian	
Scotty Adams	225-0345
Commodore Librarian	
Paul Anderson	521-3116
Education Director	
Wayne D. Schaler	728-7811
Director at Large	
Matthew Pierce	226-5155
Director at Large	
Ed Fine	829-3997
Bulletin Board	725-2312

Address all correspondence to:
Ottawa Home Computer Club,
P.O. Box 4164, Station "C",
Ottawa, Ontario K1Y 4P3.

Special Announcement

Our guest speaker this month will be Jim Butterfield, from Toronto. Well known, especially by Commodore users, Mr. Butterfield's talk is expected to be interesting to all. See the details inside this issue.

DID YOU MISS...?

by Wayne D. "SKIP" Schaler

What has the O.H.C.C., your club, been doing for the past six months?

Well, in October of 1984, several members got together to present a varied program on wordprocessing systems for Apple and Commodore machines. A special wordprocessing disk was created by Paul Anderson for the C-64. In November, our dedicated instructors presented a new and revised program of BASIC language training. In December we had several excellent presentations for our Database night. In January of 1985, forty of the top computer game programs were demonstrated by a total of 19 members, both young and old, to a delighted O.H.C.C. membership. The invitation was to bring the kids and they did. We became a family club if only for a night. In February, we returned once again to our team of professional BASIC instructors for training but added an excellent demonstration of Canada's own Telidon service from Don MacMillan.

Last month we held a languages night. Featured, among others, were COMAL, FORTH and PASCAL. Interest sub-groups already exist for COMAL and FORTH and they meet at our regular meetings. Some people have also shown an interest in PASCAL, a popular language very much in use in the business community. Machine Language was discussed that evening by Paul Anderson during his get-together session. ML will shortly become a regular class at club meetings. More information on

last month's meeting can be found in the minutes of the meeting elsewhere in this bulletin.

For April, we will have the pleasure of hearing the world renowned lecturer and Commodore Guru - Mr. Jim Butterfield - address our meeting.

=====

ABOUT JIM BUTTERFIELD

by Wayne D. "SKIP" Schaler

Actually, Mr. Butterfield needs no introduction but here is a short synopsis.

Jim lives and works out of Toronto, but travels widely. Recently, he has been to Las Vegas to demonstrate the C-264 for Commodore, to London for the annual PET Show, down south to Daytona Beach for a User-group meeting and, of course, is coming to Ottawa for our April 15 meeting.

He regularly writes articles on a wide variety of subjects for magazines such as Commodore Power Play, COMPUTE!, Transactor, TORPET, and T.P.U.G. As well, he is the author of "Machine Language for the C-64 and other Commodore Computers"; "The Commodore Reference Diary 1985"; and a new book due this June from the Robert Brady Co. called "An Introduction to Commodore Business Machine Language Programming". (Incidentally, if you have one of his books, bring it along. He might be enticed to autograph it for you, if he has the time.)

His talents are not restricted to hard copy. As well as writing the script and most of the literature, he appeared in

the much acclaimed "Bits and Bytes - The Academy" television series. This appeared first on TV Ontario in Canada and later on the Public Broadcasting System in the U.S.A.

In 1983 he produced a 90 minute videotape as an introduction to the very popular Commodore 64 personal computer.

Mr. Butterfield teaches and is widely called upon to give lectures. It is in this capacity that he is coming to Ottawa. His talk will be on advanced programming techniques and should appeal to those who have always wanted to do something more than play games with their home computers. Mr. Butterfield takes a friendly conversational approach in his highly informative lectures, making them of interest to both beginner and advanced programmers.

=====

APRIL AGENDA

by Hayne D. 'SKIP' Schaler

To cover this very special meeting properly, we have modified the agenda somewhat. There will be no sub-group meetings this evening although there will be the usual Disk of the Month sales for Apple and Commodore machines, including T.P.U.G. disks for the C-64. Extra personnel will be on hand to cover the expected increase in demand. Membership and renewal forms will also be available for those who need them. Please have the correct change to allow us to process your needs quickly.

To handle the expected larger flow of people we will start

the disk and membership sales at 7:15 P.M. and the business portion of our meeting at 7:45 SHARP. Mr. Butterfield will have the floor at 8:15 P.M. and I strongly recommend that you come early if you want seats.

The lecture will be followed by a get-together session in which members of the audience will have an opportunity to direct questions on the world of computers to Mr. Butterfield in a less formal style.

NOTE: A special thank you goes out to Mr. Don White and Mr. Brian Morrow for making Mr. Butterfield's visit to the O.H.C.C. possible.

April 15, 1985 - AGENDA

7:15 Disk of the month
Membership sales
Socialization

7:45 Business Meeting

8:15 JIM BUTTERFIELD
LECTURE

9:00 Break

9:10 Get together with JIM
BUTTERFIELD
Questions and Answers
Session

10:00 Meeting closes

★★★★★★★★★★★★

You are Invited
///

★★★★★★★★★★★★

ADAMS APPLE NOTES

by Scotty Adams



It must be time to again explain how the APPLE library works. We have no disk of the month; instead we have a library with over 200 disks that are all accessible at any time. There are eight catalogs of the complete listings of every disk at the meetings. About half of these are annotated, the rest are more or less by subject. There are specific interest disks like games, math, utilities, business, physics, art, music, etc., and many mixed interest ones. To place an order, list the disk number(s) wanted and give them to the librarian, Scotty Adams, along with the appropriate number of blank disks. They are ready for you at the next meeting for the fee of one dollar a disk. The profit from this goes to buy new disks for the library and the upkeep of the library (binders, sleeves, labels, etc.)

It would be lovely to have a better than one month turnaround, but people were calling continually to ask if their disks were ready yet and could they come over and pick them up. Since many months have more than fifty orders,

this made life a bit more complicated than usual, so I'm no longer willing to provide a speedy custom service. However, I am looking for an apprentice or assistant to help reorganize the library to better serve those with specific requests..."what do you have to teach math to a 6-year-old, what do you have on mortgages, what disk do I order for modems..." and the like.

Any disk that doesn't run on your IIc may be exchanged for another disk free...I don't have a IIc to check it on and the only way I'll know is if you tell me. If they don't run on your (ahem) clone, I'm sorry. Another thing...It's really not nice to buy disks for your non-member friends. In conclusion, I'd like to thank all the members who have contributed disks and/or documentation for the programs. Without you we would have a much poorer catalog. Thanks to you, we seem to have a bigger library than the Toronto Apple group.

=====

STRINGS IN COMAL

by Len Lindsay

*Reprinted from COMAL TODAY #5,
dated December 21, 1984*

Several people have now written asking about the missing MID\$, LEFT\$, and RIGHT\$ in COMAL. Actually, these are not missing, they are just implemented differently (better, after you understand the COMAL way). Appendix B of the COMAL HANDBOOK explains COMAL string handling. This article will only cover some important aspects.

What MID\$, LEFT\$, and RIGHT\$ are actually doing is providing your program with a substring. In COMAL, you choose the substring by naming the string followed by the characters needed specified inside parentheses:

```
a$(<start character>:<end character>)
```

If date\$="12/25/84" we can find the month, day, and year by using substrings:

```
PRINT date$(1:2) //print month
PRINT date$(4:5) //print day
PRINT date$(7:8) //print year
```

In Microsoft (Commodore) BASIC this would look more contrived:

```
PRINT MID$(D$,1,2):
REM PRINT MONTH
PRINT MID$(D$,4,5):
REM PRINT DAY
PRINT MID$(D$,7,8):
REM PRINT YEAR
```

Of course, you can use variables to choose the start and end character positions. Both start and end positions are optional. For example, to specify just the third character in the string:

```
PRINT date$(3)
```

If you want the first five characters, simply use 1 as the start:

```
PRINT date$(1:5)
```

But the best part is that you can change any part of a string without affecting the rest of it (not possible in BASIC). For instance, to change the third character from "/" into "x" we would

simply type:

```
date$(3):="x"
```

Now, date\$ is equal to "12x25/84".

This same string handling applies to string arrays too. Just specify the substring points right after the array index:

```
PRINT name$(5)(1:6)
```

This would print the first 6 characters of the 5th element of name\$ array. It is all consistent, so you also can change any part of any element of a string array.

I'll conclude with one subtle point. What if you specify a range of characters to be changed, but don't assign enough characters to it? Well, COMAL will just fill in spaces to get to the end character. For example:

```
a$="ABCDEFGG"
a$(3:5):="X"
PRINT a$
ABCX G
```

Use of this feature may not be apparent, but it does come in handy. For instance, you pad spaces on to the end of a string to make it exactly 20 characters long easily now:

```
name$(1:20):=name$
```

Did you catch that? We told COMAL to assign characters 1 thru 20 of name\$ to be equal to name\$. Now if name\$ starts out as "BOB", COMAL will add spaces to it until the 20th character. It also is easy to blank out a string variable now by turning it into all

continued on page 14

HOME COMPUTER USERS... COME FORTH!

by Ray Thomas

Why not have a look at a language which is easy to learn, is faster than BASIC, can take less memory than machine code and can be customized to your taste?

Why do I wax enthusiastic? Because FORTH is a totally unique computer language with capabilities not found in any other language.

Why fiddle with FORTH? The answer is because it is very powerful, being just as suitable for systems work (e.g. to create a hi-res display) as for applications (e.g. a personal finance program). It is about ten times faster than resident BASIC. It is extensible...you can create new commands with great ease which then become part of the language, and, for small machines, it uses little memory.

Also, it encourages the programmer to use better programming techniques and the resulting programs are portable between any machine running a standard version of FORTH.

At first sight, it has two peculiarities: it uses postfix notation (also known as reverse-polish) as used on Hewlett-Packard calculators, which means that you must write your numbers before the operators. Thus 2×4 becomes $2\ 4\ *$. This is easy to get used to and is just as logical as the more common approach. The second oddity is that a FORTH program consists of WORDS defined using lower-level words. A program, therefore, takes the form of a pyramid with

your 'superword' at the apex and the most fundamental words (primitives - written in machine code) at the base.

It would be silly of me to try to explain here how to program in FORTH when such excellent books as "Starting FORTH" by Leo Brodie may be found on the shelves of local computer stores. As well, we have in our club library a program called "Tiny FORTH Teacher" which will help to get you started.

In Ottawa, under the umbrella of the Ottawa Home Computing Club, we have a small group of enthusiastic FORTH devotees. Why not attend the informal meeting and find out more about this fascinating language?

To recap, FORTH is powerful and versatile. Most of FORTH is written in FORTH! A member of FIG once wrote PASCAL in FORTH. Many of the fast, colourful hi-res professional arcade machine games have been written in FORTH. Al Knowles has used FORTH to handle jobs usually reserved for APL.

I am a relative beginner at FORTH, but the little that I have done with it has been very enjoyable and more quickly accomplished than in BASIC or assembler.

In the beginning was the WORD!

REFERENCES:

FORTH Interest Group
P.O. Box 1105
San Carlos, CA 94070
U.S.A.

continued on page 14

=====

C-64 GAMES DESIGN

by John Batchelor

DESIGNING CRATER GUNNER

Eighth of a series

This article should appear in the March issue of OTTAWA HOME COMPUTING. By that time, I hope to have CRATER GUNNER on the Disk of the Month. During the development of the game, I kept notes on the ideas I had and on the dozens of little problems I had to solve. CRATER GUNNER puts you at the controls of a laser turret in the middle of a lunar crater. Alien saucers attack you from all sides. You hold out as long as you can. Two features make it different from simple target shooting. First, there is a scrolling background. You see less than one-sixth of the landscape at any time. You have to watch your instrument display to see if a saucer is coming from another direction. Second, the physics in this program are as accurate as possible. Your controls have some inertia or momentum. That is, they do not respond instantly. Also, the size, distance and speed of the saucers are consistent. Saucers start out small, stay small for a long time and then grow very rapidly as they close the last few hundred metres. In most games, the dumb aliens appear to slow down just as they get close! Another feature is a detailed image of the Earth in the lunar sky.

The obvious starting point is to design and implement the scrolling moon-scape. I had thought about using a circular radar scope to indicate approaching

\$5300. This decided, the next task is writing some machine code that sets up the fixed parts of the screen - the instrument panel, the border and the "stationary" parts of the crater. I use the BASIC routine at \$AB1E to do high speed prints of strings.

I realize that some of the comments in this article are going to be mysterious to those who haven't been fooling around with machine code. Sorry, see me at meetings or give me a call (226-1854) if you are sincerely interested in fuller explanations.

The next job is filling up the roughly 13K bytes of scenery! Actually, using SUPERMON (an ML monitor) you can fill memory quickly with spaces or reversed spaces. Then display blocks of memory and change a few bytes to periods (\$2E) for stars and to letters (NORTH, NNE etc.) for the compass line. The varied scenery of the crater walls is tougher but can be done by designing it right on the screen and using the T. command in SUPERMON to transfer copies of the screen to the desired locations above \$4000. This done we have the compass line of letters and reversed spaces at \$4000 to \$40FF. Spaces and stars, randomly mixed, are from \$4100 to \$4AFF. The scenery lines are \$4B00 to \$4EFF. They contain spaces, reversed spaces, left and right diagonals and a few stars. The corresponding colour data are from \$4F00 to \$52FF.

I considered adapting COLOURSCROLL from July '84 RUN to do the memory moves but it was too complex and limited to its specific demonstration. The actual machine code routine for the memory display

saucers but it took up too much room. Experimenting with character graphics, I settled on a black screen with a brown border. The top four lines are the instrument panel using reversed letters of various colours for each display. A solid yellow line separates the panel from the view screen. Ten rows of stars and the Earth sprites come next, followed by four rows of mountain peaks. The lowest row of crater wall and four rows of featureless crater floor are next. The crater rocks use the three colours of grey. The final row is in reverse light green and indicates the compass directions visible in the current screen and corresponding to the advance warning display.

Not all the rows scroll on the screen. The starry background moves and so do the crater walls. The featureless crater floor doesn't have to move. The compass line at the bottom also scrolls. Each row on the visible screen is 40 characters out of 256 permitting us to use very simple machine language to transfer to appropriate bytes onto the screen memory. We need alter colour memory only for the rows with changes in colour; that is, the crater walls. So 19 lines of 256 characters are reserved for the scrolling scenery along with four lines for scrolling colour memory. Guess it's time to figure out the memory layout of the game.

The BASIC code is at \$0800 as usual. The normal screen at \$0400 can be used too. Sprites move down from the top of video block 0 (\$3FFF down). Therefore the 23 lines of potential screen display are at \$4000 to \$52FF. Machine language starts at

was quite easy. Reserve one location in memory (I used \$CFFF) to be a pointer to the leftmost column (0-255). Then just use that pointer and the next 39 integers to index the transfer of data to screen and colour memory.

```

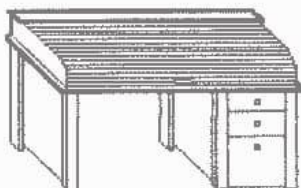
        LDX $CFFF
        LDY ##00
move LDA $4000,X
        STA $07C0,X
        INX
        INY
        CPY ##28
        BNE move

```

Of course, you would include the other lines in the move routine at the same time. This routine is the equivalent of the BASIC: $X = \text{PEEK}(49151)$; FOR Y=0 TO 39: POKE 1984+X+Y, PEEK(4*4096+X+Y): NEXT but is hundreds of times faster.

Now write a little BASIC program which changes \$CFFF in the range 0 to 255 and calls your scrolling routine repeatedly. The moonscape, sky and compass line should whiz by as \$CFFF is being incremented or decremented. That's plenty for one article. Next month, more on the development of CRATER GUNNER.

=====



BOOK REVIEW

by Sharon Ganton

=====

Title: Commodore 1541
Disk Companion

Authors: David Lawrence &
Mark England

Publishers: Sunshine Books
London, England

Price: \$16.95 W.H. Smith

=====

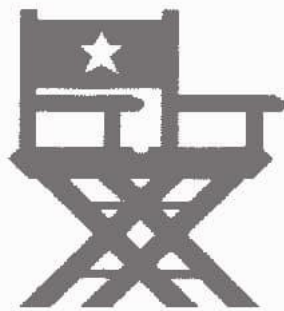
Anyone who has ever run into problems with their 1541 Disk Drive will be aware of the inadequacies of the VIC-1541 User's Manual. I understand that the newer versions of this manual have had some of the errors corrected, but after having used the Disk Companion I would never go back to my old manual.

The Disk Companion is everything my old manual is not: well written, well organized, easy to understand and with a wealth of examples. Subtitled "Secrets of the 1541 Disk Drive", this book describes in straightforward language the features and capabilities of the 1541 at a level both understandable to novice users and useful to advanced programmers.

After describing the layout of a floppy disk, the authors give some advice to beginners just setting up their systems. The next two chapters treat the Disk Housekeeping commands, with good descriptions and examples given for each one. A separate chapter discusses the error channel and includes a subroutine which can be inserted into your programs to read error messages.

The next three chapters examine Sequential and User files, Program files, and Random Files. In each case the relevant commands are explained and demonstration programs given. All programs are well documented, both within the program with REM statements and in the text with a line by line discussion. If anything, I think the authors overdo the REM statements. I found them time-consuming to type in but many GOTO and GOSUB statements jumped to a REM statement line number, so I had to be careful about omitting these comments. The program files include such goodies as a utility to merge two programs, and another to renumber any disk file. A simple database program demonstrates the handling of relative files.

Chapter Eleven takes a look at the Disk Directory and how to access it. The last chapter discusses machine code programming commands and how to execute these in the disk drive memory. An index supplements the detailed Table of Contents for ease of reference.



A Pro Looks at ProDOS - Curt Rostenbach

The WAUG - Waukegan Apple Users Group
Vol 2 No 10 October 1984

This article will discuss some of the new commands in ProDOS and some of the file types that these commands use.

File types are codes used to let you and the operating system know just what kind of information is stored within a file. Sometimes file types make important distinctions on how the data within them is to be treated and sometimes it is just for informational purposes. Both DOS 3.3 and ProDOS use file type information in their catalogs and in handling the data within files.

Under DOS 3.3, we have file types A, I, B, T, S and R. "A" is for Applesoft BASIC, "I" is for Integer BASIC, "B" is for Binary and "T" is for Text. "R" is for Relocatable binary files, which were poorly supported. "S" was never officially used by Apple Computer, maybe it was supposed to be Source, but the pirates generally used it to contain broken program code segments. And there were also another "A" and "B" file types; these were possibly for software protection but were never used (except by hackers to confuse people).

File types "A" and "I" told DOS 3.3 which BASIC had to be operating in order for the program in the file to work properly. "B" and "T" files required a little knowledge on the user's side. "Is that 'B' file a program or a Hi-res picture?", or "Is that 'T' file to be EXECed, or is it data? And if it is data, is it random access? If it is random access, what is the record size?"

ProDOS has many more file types and potential file types (up to 256!). Internally, ProDOS file types are numbers, but the catalog routine translates many of the numbers to three letter mnemonics. Under ProDOS, the major types are TXT, BIN, BAS, SYS, REL, VAR and DIR. "TXT" is for text, the same as DOS 3.3 "T" files. "BIN" is for Binary, like "B". "BAS" is for Applesoft BASIC, same as "A". "REL" is "R", the Relocatable binary files. ProDOS does not support Integer BASIC, but when and if it does, "INT" will be used like "I". ProDOS has several new file types: "DIR" for subdirectory information, "SYS" for System files, and "VAR" for Applesoft Variable files.

Since ProDOS is a near relative of Apple SOS, they share many file types - file types such as TXT and BIN. But ProDOS and SOS also have file types they do not share. They also have a great many file types that have been defined and not used, as well as file types that are reserved for future use. (See the accompanying listing of the file types.) Both ProDOS and SOS allow you to define your own file types for those special files that don't fit any of the established categories.

For the files that have been pre-established, ProDOS tries to clear up many of the ambiguities of DOS 3.3 file types. "SYS" files are for binary programs vs. "BIN" which is used for Binary data (but can still be used for programs). "TXT" files have an extra attribute that gives the record size used to create the text file. If R=0 on a "TXT" file, then the data is sequential. If R>0, then the file is a random access file with the default record size being that which is listed.

To see the files on a disk, you now have two catalog commands, "CAT" and "CATALOG". "CAT" produces a short form 40-column display, while "CATALOG" is for 80-columns and gives all the gruesome gory details of who, what, when, where and how big. (See

the listings below.) Both catalog commands can be stopped with CONTROL-C, and both give those nice to know statistics like how much space is available and how much space is used in how many files. You can specify directories and subdirectories, or use the old drive selection parameters.

To go with some of the new file types, there are now commands that create or use them. Of course, the niftiest is "-", called "dash". This is one of the most important of the lot. Didn't you get tired of "FILE TYPE MISMATCH" under DOS 3.3? I did. I hated seeing that silly thing after I typed "RUN" for a file that had to be "BRUN" or vice versa. ProDOS has an "intelligent" RUN command. Say you had a file on the disk called "GLOPS". To execute that file you could catalog the disk and see what type it is, so you can then type "EXEC GLOPS" or "RUN GLOPS" or "BRUN GLOPS". But to save a lot of bother (and typing!), "-" GLOPS would leave it up to the operating system to decide which is appropriate. In fact,

you have to use the "-" dash to get the SYS files to execute. I wish 'dash' was a little more intelligent so it could handle subdirectories as well, but there is another advantage of ProDOS over DOS 3.3 in that new commands can be added to the system- with the system's permission.

DOS 3.3 had a 'CHAIN' command to allow parts of a program to be stored separately on disk and linked together for execution. Under ProDOS, it works for Applesoft BASIC.

Applesoft BASIC programmers now have the ability to "snapshot" their variables in a program to or from a "VAR" file with the commands "STORE" and "RESTORE". What that means is that you can save your variable names AND their contents to a file, to be restored later or passed on to another program. Game programmers will make use of that feature a lot, or business programmers who want to have an "undo" option. If you want to save the current state of the variables in an Applesoft BASIC program, you have to

"STORE PATHNAME" and to return them with "RESTORE PATHNAME".

"PREFIX" is used to determine the current pathname prefix, or to set the prefix to another pathname. The pathname prefix is a string of characters that is tacked onto the beginning of a pathname (if you didn't specify a root directory) to save you a lot of typing. "PREFIX" is another intelligent command; how you use it will determine what it will do. "PREFIX" by itself will return the current prefix. If you have just booted or cataloged a disk, the prefix would be then name of the disk (ie, the "root directory"). If you used a drive specifier like "Dn" (for Drive number n), or "Sn" (for Slot number n), it will set the prefix to the root directory of the disk in that drive unit. For example, if "/GAMES" was in Drive 2, "PREFIX,D2" would set the prefix to "/GAMES". That way, to run Space Invaders you just type "-SPACE.INVADERS" instead of "-/GAMES/SPACE.INVADERS". And, if you have subdirectories, when you give a partial pathname (ie, a name that does not begin with a "/"), it appends it to the current prefix. Thus, "PREFIX ADVENTURE" would return "/GAMES/ADVENTURE" as the prefix. Finally, you can set the entire prefix by giving a root pathname, like "PREFIX /GAMES/ADVENTURE/ZORK.SERIES" and really save on the typing later. But, once again, remember that subdirectories are generally used only on hard disk drives and are not needed on floppy disks.

LISTING OF ProDOS FILE TYPES

TYPE	NAME	DESCRIPTION
\$00		Typeless file (SOS + ProDOS)
\$01	BAD	Bad block (SOS + ProDOS)
\$02		Pascal code file
\$03		Pascal text file
\$04	TXT	ASCII text (SOS + ProDOS)
\$05		Pascal data file
\$06	BIN	Binary file (SOS + ProDOS)
\$07		Font file (SOS)
\$08		Graphics screen file (SOS)
\$09		Business BASIC program (SOS)
\$0A		Business BASIC data (SOS)
\$0B		Word processor file (SOS)
\$0C		SOS system file
\$0D		Reserved for SOS
\$0E		Reserved for SOS
\$0F	DIR	Directory file (SOS + ProDOS)
\$10		RPS data file (SOS)
\$11		RPS index file (SOS)
\$12-\$BF		Reserved for SOS
\$19	ADB	Appleworks database
\$1A	AWP	Appleworks word processing
\$1B	ASP	Appleworks spreadsheet
\$C0-\$EE		Reserved for ProDOS
\$EF	PAS	ProDOS Pascal file
\$F0	CMD	ProDOS command file
\$F1-\$F8		User defined files 1-8
\$F9		Reserved for ProDOS
\$FA	INT	Integer BASIC program
\$FB		Integer BASIC variables
\$FC	BAS	Applesoft BASIC
\$FD	VAR	Applesoft variables
\$FE	REL	Relocatable (from EDASM)
\$FF	SYS	ProDOS system file

ICAT

/IAC.43

NAME	TYPE	BLOCKS	MODIFIED
*PRODOS	SYS	31	<NO DATE>
*BASIC.SYSTEM	SYS	21	<NO DATE>
*STARTUP	BAS	4	17-NOV-84
*FREWARE	DIR	1	17-NOV-84
*PRODOS.01	TXT	4	<NO DATE>
*PRODOS.02	TXT	5	<NO DATE>
*PRODOS.03	TXT	10	<NO DATE>

NOTE:

- 1) SOS refers to the operating system of the Apple ///.
- 2) This listing is incomplete; I have not include some other ProDOS files such as Pascal text, code or data because of uncertainty with their codes.

cont'd from page 7

spaces rather than just a null string:

```
name$(1:20):=""
```

For more information on strings, see the COMAL HANDBOOK appendix B.

cont'd from page 8

"Starting FORTH"
by Leo Brodie
(best book available)

"FORTH '79 Standard"
available from FIG

"BYTE" Magazine
8/80 and 4/81

"Dr. Dobb's Journal"
9/81 and 9/82

C-64 DISK OF THE MONTH

The C-64 disk is carried in the stores listed below. It is in the stores two days after each meeting.

MGI Computer Corp
1501 Carling Avenue

Compucentre
Carlingwood Mall
and Rideau Centre

Zap and Zoom
435 Kent Street

G-Plus
130 Albert Street, BB

MARCH MINUTES

by Paul Irwin

The meeting got underway at eight, when Brian Morrow welcomed one and all. After giving the latest news on the Atari front, Brian turned the floor over to Scotty Adams.

Scotty showed her slides from the San Francisco show. Lots of Macintoshes and their accessories now, and the list is growing.

Wayne Schaler introduced the Languages night by presenting the speakers. Each gave an overview of his favourite language. Following these introductions, the group broke up according to interest.

Dave Schellenburger gave a well prepared introduction to the Comal language. It runs on the C-64 and has many more friendly features to offer the BASIC programmer.

Bob Hawkins presented the Forth language with true missionary fervor. Forth is actually a religion, posing as a computer language. It ostensibly allows micros to guide telescopes and do other mundane tasks.

Paul Irwin introduced the Pascal language. Pascal is a professional language, used on minis and mainframes a lot. It is somewhat popular on Apples, but really needs a hard disk to provide a decent tool, for the hardcore micro buff.

Apple and Commodore groups also met briefly. Scotty gave a demo of mouse graphics on the //e.

All in all, a busy and interesting meeting.

Join **TPUG** the World's largest Commodore users' group



- Free information-packed **Monthly Magazine**
- Access to 4000 **Public Domain Programs**
- Annual **Convention**
- Member **Bulletin Board**
- For \$30.00 (\$20.00 for students), all the above and attend the many TPUG meetings in the **Toronto Area**

* * * * *

SPECIAL RATE FOR Ottawa 6502 members!!!

Contact Don White for details.

* * * * *

MACCOM SYSTEMS LTD.



	3½ MAC \$59.00	
MEMOREX ssdd \$24.95	XIDEX ssdd \$29.99	DYSAN ssdd \$32.99

in stock
now at

43 WILLIAM ST.
749-1232

PaperClip	\$84.95	Flight Sim2	64.99
BusCard II	169.95	Koala Pad	\$99.99
BI 80	\$169.95	CODE Pro	\$77.00

Astro Positions



ASTRO POSITIONS

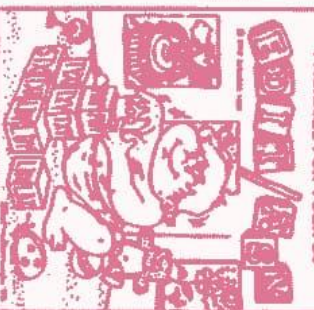
display and print the position of the sun and stars (one arc second accuracy) and the position of the moon and the planets (one arc minute accuracy except for Pluto) for any date in history.

includes User's Guide, *Introduction to Positional Astronomy and Introduction to Classical Astrology*.

DISK 33.95 US 42.95 CAN

Software for Commodore-64

COMPILED BY EDITOR, ASSEMBLER, PACKAGER FOR YOUR 64



EDITOR/ASM II

- DESIGNED TO HELP YOU CREATE AND MODIFY 6502 Assembly Language Programs on the Commodore 64 Computer

ON SCREEN HELP SYSTEM to assist you in using all commands available; add, assemble, calculate, delete, directory, disk, edit, find, help, insert, list, load, merge, new, plot, quit, and save.

DISK 28.95 US 36.95 CAN



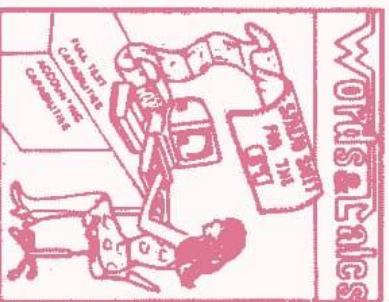
DAISY

Database Information System

one of the comprehensive DATA-BASE SYSTEMS for the 64

fully bilingual on screen calculates, compute the contents of numeric fields - add, subtract, multiply or divide against the defined field using either constant value of the contents of any other field in the record. Virtual disk operation.

DISK 31.95 US 39.95 CAN



WORDS & CALC

Powerful household finance -

Business system eliminate inconvenient files, confusing tables and tedious calculations store all your data on WORDS & CALCs spreadsheet WORDS & CALCs will do your work for you in organizing, displaying, storing, printing and performing calculations

DISK 33.95 US 42.95 CAN

YES! You CAN get quality software

at a reasonable price.

Write for our FREE catalog.

Money Order, Check, Visa or Mastercard. Dealer and distributor inquiries invited.



Suite 210,
5950 Côte des neiges
Montreal, Quebec H3S 1Z6
Canada